



Öğr. Gör. Mustafa GÖKMEN
gokmen@selcuk.edu.tr

Nesneye Yönelik Programlama

Flow Control Statements

Kontrol İfadeleri (Control Statements)



- ✓ Kontrol ifadeleri, bir takım koşullara göre programın akışının belirlenmesini sağlayan programlama karşılaştırma ifadeleridir. Bir kontrol ifadesi, işletilecek bir sonraki komutun hangisi olacağını belirlemek üzere bir koşulu denetler.
- ✓ Kontrol ifadeleri belirli bir koşula göre farklı işlemler yaptırılacağı zaman kullanılır.
 - Örneğin not adlı değişken değeri 5'ten küçük olduğunda farklı, 5'ten büyük veya eşit olduğunda farklı kodlar yürütülecekse bu işi yapabilmek için karşılaştırma deyimlerine gereksinim duyulur.
- ✓ Java'da if-else ve switch-case olmak üzere iki farklı kontrol ifadesi vardır.

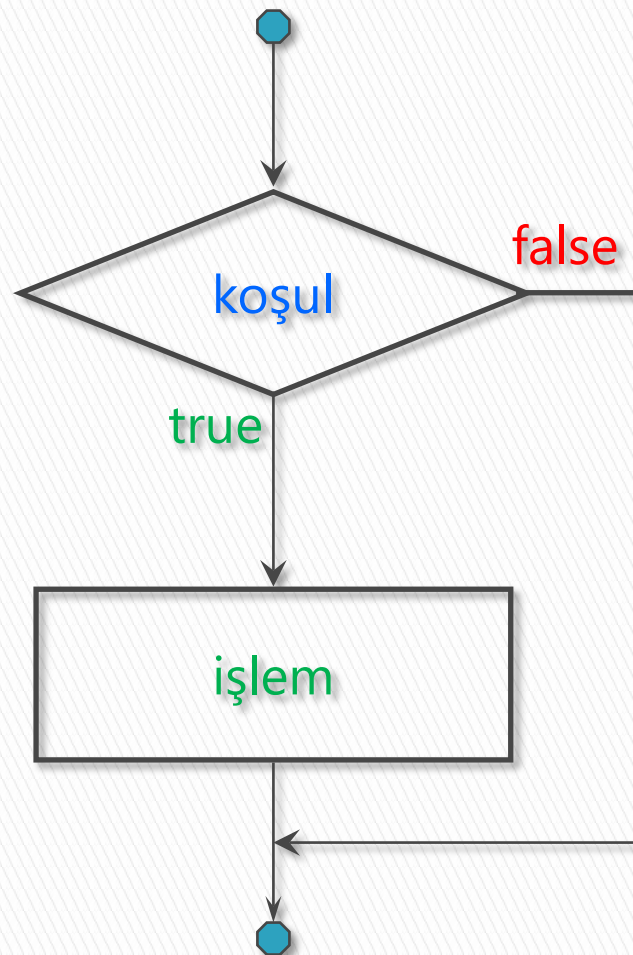
Basit if deyimi

Kullanım şekli:

```
if ( koşul ) {  
    .  
    işlem;  
    .  
}
```

Eğer koşul doğruysa işlem bloku çalıştırılır, değil ise her hangi bir işlem yapılmadan devam edilir.

- Yapılacak işlem, tek bir komut veya birden fazla komuttan oluşabilir.



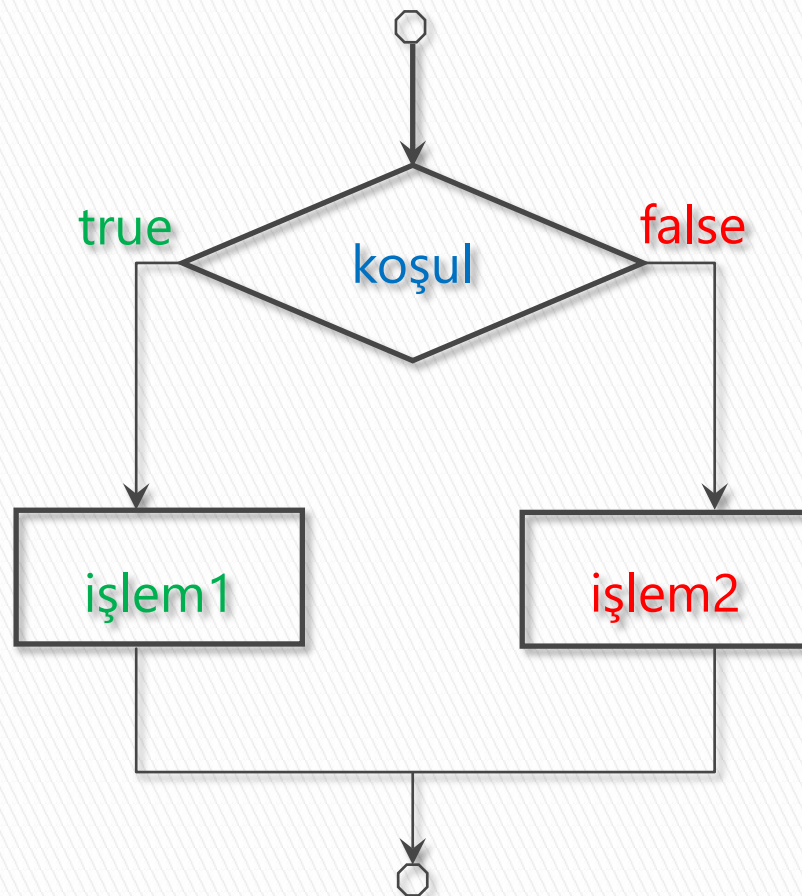
if-else deyimi

Kullanım şekli;

```
if ( koşul ) {  
    işlem1  
} else {  
    işlem2  
}
```

Eğer koşul doğru ise **işlem1**, değil (else) ise **işlem2** çalıştırılacaktır.

```
if (sayi == 0) {  
    System.out.println("sayi 0'dir.");  
} else {  
    System.out.println("sayi 0 değildir.");  
}
```



if-else deyimi

```
if ( <karşılaştırma ifadesi> )  
    <doğru ise çalıştırılacak kod>  
else  
    <yanlış ise çalıştırılacak kod>
```

```
Scanner sc = new Scanner(System.in);  
System.out.print("İki sayi giriniz?: ");  
  
int a = sc.nextInt();  
int b = sc.nextInt();  
  
if (a < b)  
    System.out.print("a sayısı b sayısından küçük");  
else  
    System.out.print("a sayısı b sayısından küçük degil.");  
sc.close();
```

İki sayi giriniz?: 5 8
a sayısı b sayısından küçük

İki sayi giriniz?: 8 5
a sayısı b sayısından küçük degil.

if-else deyimi

```
public class Comparison01 {  
    public static void main(String[] args) {  
        int a = 0, b = 10;  
        boolean x1, x2, x, y1, y2, y;  
        x1 = (a <= 5);    // x1 -> true  
        x2 = (b > 10);    // x2 -> false  
        x  = (x1 && x2);  // x1 && x2 = true && false = false  
        y1 = (a < 1);     // y1 -> true  
        y2 = (b >= 9);    // y2 -> false  
        y  = (y1 || y2);  // y1 || y2 = true || false = true  
        if (x && y) {     // x && y = false && true = false  
            System.out.println("Kosul dogru");  
        } else {  
            System.out.println("Kosul yanlis");  
        }  
    }  
}
```

Kosul Yanlis

```
public class Comparison02 {  
    public static void main(String[] args) {  
        int a = 0, b = 10;  
        if (((a <= 5) && (b > 10)) && ((a < 1) || (b >= 9))) {  
            System.out.println("Kosul dogru");  
        } else {  
            System.out.println("Kosul yanlis");  
        }  
    }  
}
```

if-else deyimi

Operatörlerin önceliği

```
public class Comparison03 {  
    public static void main(String[] args) {  
        int a = 0, b = 10;  
        if ((a <= 5) && (b > 10) && (a < 1) || (b >= 9)) {  
            System.out.println("Kosul dogru");  
        } else {  
            System.out.println("Kosul yanlis");  
        }  
    }  
}
```

Kosul Dogru

Eksik parantezlere dikkat!..
Kastedilen şeye göre hatalı da olabilir, doğru da olabilir.
Sorumluluk programcıdadır.

if-else deyimi

```
import java.util.Scanner;
public class IfElse {
    static Scanner kb = new Scanner(System.in);
    public static void main(String[] args) {
        System.out.println("Alan Hesaplama: 0-Daire 1-Dortgen");
        System.out.print("Seciminiz? (0-1): ");
        int secim = kb.nextInt();
        if (secim == 0) {
            System.out.print("Yaricap:");
            int yr = kb.nextInt();
            System.out.println("Dairenin Alani: " + yr * yr * Math.PI);
        } else {
            System.out.print("Birinci Kenar: ");
            int kenar1 = kb.nextInt();
            System.out.print("Ikinci Kenar: ");
            int kenar2 = kb.nextInt();
            int Alan = kenar1 * kenar2;
            System.out.println("Dortgen Alani: " + Alan);
        }
    }
}
```

Alan Hesaplama: 0-Daire 1-Dortgen
Seciminiz? (0-1): 0
Yaricap: 5
Dairenin Alani: 78.53981633974483

Alan Hesaplama: 0-Daire 1-Dortgen
Seciminiz? (0-1): 1
Birinci Kenar: 5
Ikinci Kenar: 6
Dortgen Alani: 30

if-else deyimi

Bir koşulun sağlanmadığı durumda diğer koşul ya da koşulların denetlenmesini sağlayan bir yapıdır. Denetlenen koşullardan herhangi birisinin sonucu mantıksal doğru olursa, o kod bloğu işletilir ve programın akışı en sonraki **else** bloğunun bitimine sapar. Hiç bir koşul doğru değilse, o zaman **else** bloğu işletilir.

```
if (1. kosul ifadesi) {  
    1. kosul dogruysa isletilecek kod bloku  
} else if (2. kosul ifadesi) {  
    2. kosul dogruysa isletilecek kod bloku  
.  
. // else if bloku gerektiği kadar tekrarlanabilir  
.  
} else {  
    Hic bir kosul dogru degilse isletilecek kod bloku  
}
```

if-else if-else deyimi

```
public class IfElseIfElse {  
    public static void main(String[] args) {  
        int puan = 81;  
        char karne;  
  
        if (puan >= 90) {  
            karne = 'A';  
        } else if (puan >= 80) {  
            karne = 'B';  
        } else if (puan >= 70) {  
            karne = 'C';  
        } else if (puan >= 60) {  
            karne = 'D';  
        } else {  
            karne = 'F';  
        }  
  
        System.out.println("Harf Notu = " + karne);  
    }  
}
```

Harf Notu = B

if-else if-else deyimi

```
import java.util.Scanner;
public class IfMultiChoice {
    static Scanner kb = new Scanner(System.in);
    public static double alanDaire() {
        System.out.print("Yaricap: ");
        Double param = kb.nextDouble();
        return Math.pow(param, 2) * Math.PI;
    }
    public static double alanDortgen(float par1, float par2) {
        return par1 * par2;
    }
    public static void main(String[] args) {
        System.out.println("Alan Hesaplama: 0-Daire 1-Dörtgen");
        System.out.print("Seciminiz? (0-1): ");
        Integer secim = kb.nextInt();
        kb.reset();
        if (secim == 0) {
            System.out.printf("Dairenin Alani: %3.3f\n", alanDaire());
        } else if (secim == 1) {
            System.out.print("Kenarlar: ");
            Float k1 = kb.nextFloat();
            Float k2 = kb.nextFloat();
            System.out.printf("Dortgenin Alani: %3.1f\n", alanDortgen(k1, k2));
        } else {
            System.out.printf("Hatali Secim: %s", secim);
        }
    }
}
```

Dikkat!..

- 1-) Yeni Scanner nesnesinin oluşturulduğu yer, oluşturulma şekli ve kapatılmaması
- 2-) if - else if - else bloklarının kullanımı
- 3-) Method çağrıları ve parametre geçişi
- 4-) Dönüş değerlerinin kullanımı
- 5-) Sarmalayıcı (Wrapper) veri türlerinin kullanımı
- 6-) String ifadelerin biçimlendirmesi (printf string formatı)
- 7-) Matematik fonksiyonların (pow() gibi) kullanımı

```
Alan Hesaplama: 0-Daire 1-Dörtgen
Seciminiz? (0-1): 0
Yaricap: 5,1
Dairenin Alani: 81,713
```

```
Alan Hesaplama: 0-Daire 1-Dörtgen
Seciminiz? (0-1): 1
Kenarlar: 1,1 2,2
Dortgenin Alani: 2,4
```

switch-case deyimi

switch deyimi, **if-else if-else** deyimleri ile yazılabilecek bir kod kesimini daha basitçe yazmayı sağlayan, bir ifadenin değerine göre dallanmayı sağlayabilen bir deyimdir.

```
switch ( ifade ) {  
    case deger1:  
        kod bloku;  
        break;  
    case deger2:  
        kod bloku;  
        break;  
    case degerN:  
        kod bloku;  
        break;  
    default:  
        kod bloku;  
}
```

switch-case deyimi

Not ortalamamız nedir?: 88
Harf notunuz B+'dır.

```
import java.util.Scanner;
public class SwitchCase {
    static Scanner kb = new Scanner(System.in);
    public static void main(String[] args) {
        String karne;
        System.out.print("Not ortalamamız nedir?: ");
        int puan = kb.nextInt();
        switch (puan / 10) {
            case 10: karne = "A"; break;
            case 9: karne = "A-"; break;
            case 8: karne = "B+"; break;
            case 7: karne = "B"; break;
            case 6: karne = "B-"; break;
            case 5: karne = "C+"; break;
            case 4: karne = "C"; break;
            case 3: karne = "C-"; break;
            case 2: karne = "D+"; break;
            case 1: karne = "D"; break;
            case 0: karne = "F"; break;
            default: karne = "Yanlis not";
        };
        if (!karne.isEmpty() && ((puan >= 0) && (puan <= 100))) {
            System.out.printf("Harf notunuz %s'dir.", karne);
        } else {
            System.out.println("Yanlis not girdiniz.");
        }
    }
}
```




Öğr.Gör. Mustafa GÖKMEN
gokmen@selcuk.edu.tr

Nesneye Yönelik Programlama

Java
Loops

Loops (Döngüler)

- Döngüler, bir koşul doğru olduğu sürece verilen komutların tekrar tekrar çalıştırılması için kullanılan ifadelerdir.
- Her hangi bir andaki koşul doğru olduğu sürece ilgili kod bloku veya komut satırı çalıştırılır.
- Koşulun yanlış olduğu anlaşıldığında ise normal olarak veya özel durumlarda "break" ile döngüden çıkılabilir.
- Eğer bir koşul için ilgili komut çalıştırılmak istenmez ise "continue ifadesi" kullanılarak döngüye sonraki koşul denetimi ile devam edilebilir.

while döngüsü

while sözcüğü ile koşulun içine yazıldığı parantezler zorunludur. Koşul sağlandığı sürece **while** bloğunun içindeki kod kesimi işletilecektir.

```
public class WhileLoop {  
    public static void main(String[] args) {  
        int n = 6, factorial = 1;  
        while (n > 1) {  
            factorial *= n--;  
        }  
        System.out.println("6! = " + factorial);  
    }  
}
```

Başlangıç Değerleri:
n=6, factorial=1

n > 1	n	factorial	n * factorial
true	6	1	6
true	5	6	30
true	4	30	120
true	3	120	360
true	2	360	720
false	n = 1 (döngüden çıkıldı)		

do-while döngüsü

Bu döngü deyiminde koşul sınaması döngü sonunda yapılır. Dolayısıyla çevrim kod öbeği en az bir kez yürütülmüş olur.

```
public class WhileDoLoop {  
    public static void main(String[] args) {  
        int n = 1;  
        int factorial = 1;  
        do {  
            factorial *= n++;  
        } while (n <= 6);  
        System.out.println("6! = " + factorial);  
    }  
}
```

Başlangıç Değerleri:
n=1, factorial=1

n	factorial	factorial*n	n <= 6
1	1	1	true
2	1	2	true
3	2	6	true
4	6	24	true
5	24	120	true
6	120	720	false

for döngüsü

Bu döngü deyiminde koşulda kullanılan çevrim değişkeni, koşul ifadesi ve çevrim sayacı artımı for ifadesi içerisinde verilir.

```
for (başlangic degerleri; kosullar; kontrol ifadeleri) {  
    <dongu kod bloku>  
}
```

Başlangıç Değerleri:
n=1, factorial=1

```
public class ForLoop01 {  
    public static void main(String[] args) {  
        int n = 1;  
        int factorial = 1;  
        for (n = 1; n <= 6; n++) {  
            factorial *= n;  
        }  
        System.out.println("6! = " + factorial);  
    }  
}
```

n	factorial	factorial*n	n <= 6
1	1	1	true
2	1	2	true
3	2	6	true
4	6	24	true
5	24	120	true
6	120	720	False
7	n = 7 (Döngüden çıkıldı)		

for döngüsü (Farklı kullanım)

Başlangıç ve kontrol deyimleri birden fazla tanımlanabilir ancak dikkatli bir şekilde kodlanması gereklidir.

```
public class ForLoopMulti {  
    public static void main(String[] args) {  
        int i, j;  
  
        for (i = 1, j = 9; ((i <= 9) && (j >= 1)); i++, j--) {  
            System.out.printf("%d + %d = %d\n", i, j, i + j);  
        }  
    }  
}
```

```
1 + 9 = 10  
2 + 8 = 10  
3 + 7 = 10  
4 + 6 = 10  
5 + 5 = 10  
6 + 4 = 10  
7 + 3 = 10  
8 + 2 = 10  
9 + 1 = 10
```

```
public class ForLoopMulti {  
    public static void main(String[] args) {  
        int i, j;  
  
        for (i = 1, j = 9; ((i < 9) && (j > 1)); i++, j--) {  
            System.out.printf("%d + %d = %d\n", i, j, i + j);  
        }  
    }  
}
```

```
1 + 9 = 10  
2 + 8 = 10  
3 + 7 = 10  
4 + 6 = 10  
5 + 5 = 10  
6 + 4 = 10  
7 + 3 = 10  
8 + 2 = 10
```

Gelişmiş for döngüsü

for-each stili bir döngü, dizi gibi bir nesne koleksiyonu üzerinde kesin sıralı bir biçimde baştan sona kadar döngü oluşturmak için tasarlanmıştır. for döngüsünün for-each stili aynı zamanda gelişmiş for döngüsü olarak da bilinir. Genel biçimi aşağıdaki şekildedir.

```
for (tip degisken: Koleksiyon) {  
    degiskeni kullanan kod bloku  
}
```

Burada **tip** baştan sona dek, her seferinde bir tane olmak üzere **Koleksiyon**dan elemanları birer birer alacak olan iterasyona ait değişken tipi ve **degisken** ise bu değişkenin adıdır. Döngü kurulan koleksiyon, **Koleksiyon** ile belirtilir. Döngünün her iterasyonunda, koleksiyonun sıradaki eleman alınır ve **degisken** içinde depolanır. Döngü, koleksiyondaki tüm elemanlar elde edilene kadar tekrarlanır.

Gelişmiş for döngüsü

```
public class ForEach01 {  
    public static void main(String[] args) {  
        Integer[] sayilar = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };  
        for (Integer i : sayilar) {  
            System.out.printf("%2d^2 = %3.0f\n", i, Math.pow(i, 2));  
        } // Diziler konusunda daha fazlasını göreceğiz.  
    }  
}
```

1^2	=	1
2^2	=	4
3^2	=	9
4^2	=	16
5^2	=	25
6^2	=	36
7^2	=	49
8^2	=	64
9^2	=	81
10^2	=	100

```
public class ForEach02 {  
    public static int maksimum(int[] sayilar) {  
        int enbuyuk = sayilar[0];  
        for (int s : sayilar) {  
            if (s > enbuyuk) {  
                enbuyuk = s;  
            }  
        } // Diziler konusunda daha fazlasını göreceğiz.  
        return enbuyuk;  
    }  
}
```

En yüksek ücret: 995

```
public static void main(String[] args) {  
    int[] ucet = { 850, 970, 615, 550, 488, 827, 995, 108 };  
    int max = maksimum(ucet);  
    System.out.println("En yuksek ucet: " + max);  
}
```

Sonsuz Döngü (Continuous Loop)

`for` ifadesinde başlangı, koşul ve kontrol deyimleri verilmez ise bu döngü sürekli döngü olarak adlandırılır ve sona erdirmek için kod düzenlenmelidir. Aynı işlem `while(true)` ile de yapılabilir.

```
import java.util.Scanner;

public class ForContLoop {
    static Scanner kb = new Scanner(System.in);

    public static void main(String[] args) {
        int carpim = 1, sayi = 0;
        for (;;) {
            System.out.print("Sayi? (0=Son): ");
            sayi = kb.nextInt();

            if (sayi == 0) {
                break;
            } else if (sayi == 1) {
                continue;
            }
            carpim *= sayi;
        }
        System.out.println("Carpim: " + carpim);
    }
}
```

```
import java.util.Scanner;

public class WhileContLoop {
    static Scanner kb = new Scanner(System.in);

    public static void main(String[] args) {
        int carpim = 1, sayi = 0;
        while (true) {
            System.out.print("Sayi? (0=Son): ");
            sayi = kb.nextInt();

            if (sayi == 0) {
                break;
            } else if (sayi == 1) {
                continue;
            }
            carpim *= sayi;
        }
        System.out.println("Carpim: " + carpim);
    }
}
```